BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

Attorney's Docket No.: 442-009325-US(PAR) *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Express Mail No.: EL336863125US
In re application of: LINDEN et al.
Serial No.: 0 /
Filed: Herewith
For: NETWORK ELEMENT

Group No.:

Examiner:

Commissioner of Patents and Trademarks
Washington, D.C. 20231

TRANSMITTAL OF CERTIFIED COPY

Attached please find the certified copy of the foreign application from which priority is claimed for this case:

Country	**: Finland**
Application Number	**: 990800**
Filing Date	**: 12 April 1999**

WARNING: "When a document that is required by statute to be certified must be filed, a copy, including a photocopy or facsimile transmission of the certification is not acceptable." 37 CFR 1.4(f) (emphasis added.)

SIGNATURE OF ATTORNEY

Reg. No.: 24,622

Clarence A. Green

Type or print name of attorney

Tel. No.: (203) 259-1800

Perman & Green, LLP

P.O. Address

425 Post Road, Fairfield, CT 06430

NOTE: The claim to priority need be in no special form and may be made by the attorney or agent if the foreign application is referred to in the oath or declaration as required by § 1.63.

PATENTTI- JA REKISTERIHALLITUS
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 25.11.1999

E T U O I K E U S T O D I S T U S
P R I O R I T Y D O C U M E N T

Hakija / Applicant: **Nokia Mobile Phones Ltd**
Espoo

Patenttihakemus nro / Patent application no: **990800**

Tekemispäivä / Filing date: **12.04.1999**

Kansainvälinen luokka / International class: **H04L**

Keksinnön nimitys / Title of invention:

"Verkkoelementti"

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.

Pirjo Kalla
Tutkimussihteeri

Maksu 300,- mk
Fee 300,- FIM

Osoite: Arkadiankatu 6 A, P.O.Box 1160, FIN-00101 Helsinki, FINLAND
Puhelin: 09 6939 500
Telephone: + 358 9 6939 500
Telefax: 09 6939 5204
Telefax: + 358 9 6939 5204

Verkkoelementti

Esillä oleva keksintö koskee tiedonsiirtoa ja erityisesti tiedonsiirtoverkon verkkoelementti, joka käsittää tiedonsiirtovälineet tiedon lähettämiseksi ja vastaanottamiseksi tiedonsiirtoverkosta, joka tieto käsittää yhden tai useampia suorituskäskyjä; käsittelyvälineet määrätyssä muodossa annetun tiedon käsittelemiseksi; ja ohjausvälineet vastaanotetun suorituskäskyn muokkaamiseksi käsittelyvälineiden edellyttämään muotoon.

Ajuriksi (englanniksi:driver) kutsutaan elektronisten laitteiden yhteydessä yleisesti toiminnallista yksikköä tai kriteeriä, joka käyttää tai ohjaa toista järjestelmää tai komponenttia. Tyypillisesti ajuri muodostuu ohjelmistomoduulista, joka sisältää tarvittavat toiminnallisuudet laitteistoyksikön kytkemiseksi järjestelmään määrätyn laitteistorajapinnan välityksellä. Tämä tarkoittaa etupäässä laitteistoyksikköön liittyvää ohjelmakokonaisuutta, joilla järjestelmän antamat käskyt tulkitaan ja muokataan laitteistoyksikön ymmärtämään muotoon ja päinvastoin.

Langattomien päätelaitteiden käytön laajentuessa ääniviestinnästä muihin mediamuotoihin ja datapalveluihin, ajureiden tarve päätelaitteeseen lisättävien toimintojen tukemiseksi kasvaa. Langattomille päätelaitteille on tyypillistä tavanomaisia päätelaitteita pienempi muistikapasiteetti ja suorituskyky, ja tämä vaikuttaa myös ajureiden järjestämiseen mainitussa ympäristössä.

Eräs esimerkki nopeasti kasvaneesta kehityssuunnasta on langattoman päätelaitteen avulla käytettävät älykorttisovellukset. Älykortin käyttöä varten päätelaitteeseen on liitetty kortinlukija, johon käyttäjä palvelutapahtuman yhteydessä asettaa älykortin. Älykortin käytön hallintaan liittyvät komennot käyttäjä antaa päätelaitteen käyttöliittymän avulla. Päätelaite hoitaa tarvittaessa yhteyden sovellusta ylläpitävän palvelimeen ja käsittää ajurin, joka ohjaa palvelutapahtumaan liittyviä toimintoja sekä palvelimen että kortin suuntaan.

Yleistyneitä älykorttisovelluksia ovat esimerkiksi elektroniset kukkarot, joihin liittyviä palveluntarjoajia on jo useampia. Tyypillisesti kullakin kukkarosovelluksella on oma matalan tason rajapintansa, joka on vielä erilainen sen mukaan mistä kukkarosovelluksesta on kyse tai minkä tyyppinen korttiyhdistelmä on kyseessä. Yhdellä kortilla voi olla myös useampia älykorttisovelluksia, kuten esimerkiksi asiakaspalkkiosovellus, käytönseurantasovellus jne. Jo nyt voidaan nähdä, että sovellusten ja laitevalmistajien määrän kasvaessa erillisen ajurin järjestäminen monille eri vaihtoehdoille tulee aiheuttamaan ongelmia langattomassa päätelaitteessa tarvittavan ohjelmiston ja päätelaitteen muistintarpeen osalta.

Yleensä ajuri järjestetään langattomaan päätelaitteeseen konekielisenä koodina, jonka laitevalmistaja sisällyttää päätelaitteen ohjelmistoon jo tuotantovaiheessa. Tämä tarkoittaa sitä, että päätökset siitä mitä älykorttisovelluksia tuetaan tehdään jo varhaisessa vaiheessa ja oleellisesti laitevalmistajan toimesta. Eri laitevalmistajat voivat tukea eri sovelluksia tai jopa saman laitevalmistajan eri mallisukupolvet voivat tukea eri sovelluksia, mikä lisää tuotevalikoiman kirjavuutta ja vaikeuttaa käyttäjien päätöksiä vaihtoehtoja tehtäessä. Päätelaitteen ohjelmisto on altis vanhentumiselle, ja ohjelmistoversioiden ylläpitäminen vaatii lisätyötä palvelun kaikilta osapuolilta. Joissakin kehittyneemmissä malleissa (esimerkiksi Nokia 9110 Communicator) on mahdollisuus ladata konekielistä koodia jälkikäteen päätelaitteen muistiin, mutta tällaisessa ratkaisussa päädyttäisiin vastaavasti ylläpitämään tietoa kunkin laitevalmistajan tavasta järjestää koodin lataus jälkikäteen. Uuden palveluntarjoajan kannalta tekniikan tason mukainen järjestely on hankala, sillä palveluntarjoaja joutuu varsinaisen asiakasmyynnin lisäksi huolehtimaan, että mahdollisimman monet laitevalmistajat ovat halukkaita tukemaan kyseistä palvelua.

Nyt on keksitty ratkaisu, jonka avulla voidaan muodostaa yhteys verkkoelementin ja verkkoelementissä toteutettavien sovellusten välille joustavalla ja silti turvallisuuden ylläpitämisen edellyttämällä tavalla. Eräs edullinen suoritusmuoto on tuen aikaansaaminen uusille älykorttisovelluksille langattomassa päätelaitteessa tai siihen toiminnallisesti yhdistettävissä olevan lisälaitteen avulla.

Keksinnön mukaisessa ratkaisussa verkkoelementin ohjausvälineet käsittävät ajurin, jonka alkuperä on varmennettavissa elektronisen allekirjoituksen avulla; ja yhden tai useampia käsittelyvälineiden toimintaa ohjaavia funktioita, jotka ovat käynnistettävissä vain ajurilla, jonka alkuperä on varmennettu elektronisen allekirjoituksen avulla.

Keksinnössä hyödynnetään uusien järjestelmäalustojen tarjoamaa mahdollisuutta siirtää suurempia määriä tietoa, erityisesti ohjelmakoodia, radiorajapinnan yli. Keksinnön mukaisessa ratkaisussa tiettyyn sovellukseen liittyvä ajuri järjestetään edullisesti palvelimelle, josta se on ladattavissa sovellusta toteuttavaan verkkoelementtiin. Ajuri toteutetaan tavukäännettynä koodina (byte compiled code) ja tallennetaan verkkoelementtiin (esimerkiksi päätelaitteen haihtumattomaan muistiin, edullisesti flash-muistiin). Ajuri tarjoaa vakiomuotoisen, ylemmän tason sovellusrajapinnan (High Level Application Programming Interface, HL-API) verkkoelementissä toteutettaville sovelluskoodeille, jotka käyttövaiheessa edullisesti ladataan verkkoelementtiin. Älykorttiratkaisun yhteydessä tiedonsiirrossa älykortin suuntaan ajuri käyttää sovelluskohtaisen yhteyskäytännön mukaisia tietoyksiköitä (Application Protocol Data Unit, APDU). Päätelaitteelle on tallennettu, edullisesti konekielisenä, koodikirjasto, jonka funktioita ajuri kutsuu tiedonsiirtotapahtumien yhteydessä antaen funktiolle parametrinä kulloinkin tarvittavan APDUn .

Turvallisuuden ylläpitämiseksi edullisesti ajurin julkaisija varustaa ajurin elektronisella allekirjoituksella, jonka avulla voidaan varmentaa se, että ladattava tavukäännetty koodi on nimenomaan allekirjoittajan julkaisema ohjelmisto. Elektroninen allekirjoitus voidaan tarkistaa ajurin latausvaiheessa, tai haluttaessa esimerkiksi ajurin käynnistyksen yhteydessä. Jos ajurin alkuperää ei voida elektronisen allekirjoituksen avulla varmentaa, ajuri hylätään. Keksinnön mukaisessa ratkaisussa verkkoelementtiin edullisesti tuotantovaiheessa kiinteästi tallennettu koodikirjasto käsittää ainakin kaksi osaa, vakiokirjaston ja suljetun kirjaston. Vakiokirjasto sisältää yhden tai useampia funktioita, joita voidaan kutsua mistä tahansa sovelluskoodista käsin. Suljetun kirjaston funktioita sen sijaan voi

hyödyntää vain sovelluskoodi, jonka elektroninen allekirjoitus on hyväksyttävästi tarkistettu.

Keksintö helpottaa uusien sovellusten tukemista verkkoelementeissä, esimerkiksi älykorttisovellusten tuen järjestämistä liikkuviin päätelaitteisiin. Ajuri on edullisesti ladattavissa päätelaitteeseen ilmarajapinnan yli jolloin tavukäännetyn koodin ja vakiomuotoisen alemman tason rajapinnan ansiosta sama ajuri soveltuu käytettäväksi erilaisissa päätelaitteissa, valmistajasta tai edes päätelaitetyypistä riippumatta. Mahdollisuus käyttää päätelaitteeseen tallennettua ajuria, joka tarjoaa vakiomuotoisen ylemmän tason rajapinnan pienentää sovelluskoodin kokoa ja siten samalla yhteysaikaa palvelutapahtumien yhteydessä. Ajuri voi itsessään olla suhteellisen suurikokoinen, mutta koska se on tallennettavissa päätelaitteelle, lataustoimenpide joudutaan tekemään harvoin. Sovellusohjelmoijan ei tarvitse huolehtia siitä miten sovellukset toteutetaan alemmalla tasolla tai miten toteutukset poikkeavat toisistaan, tai välttämättä edes minkälaisella päätelaitteella tai päätelaitteen lisälaitteella sovellus on tarkoitus toteuttaa.

Keksinnön mukainen ratkaisu on turvallinen, sillä potentiaalisesti turvallisuutta vaarantavat toiminnot keskittyvät ajuriin, jonka julkaisija on aina todennettavissa elektronisen allekirjoituksen avulla. Lisäksi myös niiden sovellusten toteutus, jossa ei tarvita suljetun kirjaston tietoyksiköiden avulla toteutettavia toiminnallisuuksia, hyötyy aikaansaadusta joustavuudesta. Organisaatioille, joille sovellusten tarjoaminen ei perinteisesti ole turvallisuusnäkökohtien valossa kovinkaan helppoa, tuo keksinnön tarjoama mahdollisuus sovellusten alkuperän varmistamiseksi käytettävän ajurin allekirjoituksen perustella helpomman keinon turvallisuusvaatimuksiltaan vaativampien sovellusten aikaansaamiseksi.

Keksintöä selostetaan seuraavassa yksityiskohtaisesti viittaamalla oheisiin piirustuksiin, joissa:

kuviossa 1 on esitetty periaatekuva WAP-mallista (Wireless Application Protocol);

kuvion 2 lohkokaavion avulla on havainnollistettu asiakkaana käytettävää langatonta päätelaitetta;

kuviossa 3 tarkastellaan periaatetasolla keksinnön mukaista älykorttisovelluksen toteutusta toiminnallisuuksien mukaisesti esitettynä;
kuviossa 4 on esitetty tarkemmin keksinnön mukainen järjestely älykorttisovelluksen toteuttamiseksi langattomassa päätelaitteessa;
kuviossa 5 on esitetty eräs keksinnön suoritusmuoto; ja
kuviossa 6 on havainnollistettu järjestely yhteen käyttötarkoitukseen tarkoitettujen sovellusten toteuttamiseksi päätelaitteella.

WAP (Wireless Application Protocol) on WAP Forum -yhteenliittymän määrittämä järjestely Internetin ja kehittyneiden datapalveluiden saannin toteuttamiseksi langattomissa päätelaitteissa. WAP tarjoaa periaatteessa skaalautuvan ja laajennettavissa olevan kokonaisuuden, jonka kerrosrakenteisessa arkkitehtuurissa tietty yhteyskäytäntökerros tarjoaa palveluita seuraavalle kerrokselle. WAP-arkkitehtuuri on hyvin lähellä Internetistä tunnettua WWW-mallia, mutta siihen on tehty langattoman ympäristön vaatimia optimointeja ja muutoksia. Kuviossa 1 on esitetty periaatekuva WAP-mallista, joka mahdollistaa asiakkaan ja palvelimen välisen neuvottelun palvelimella säilytettävän dataobjektin tarjoamiseksi lukijalle ymmärrettävässä muodossa. Asiakas 1 lähettää langattoman verkon 2 yli yhdyskäytävälle 3 koodatun palvelupyynnön, jonka yhdyskäytävä 3 dekoodaa ja välittää Internetin 4 välityksellä palvelimelle 5. Palvelin 5 lähettää pyydetyn sisällön yhdyskäytävälle 3, joka koodaa sisällön ja lähettää sen palvelupyynnön tehneelle asiakkaalle 1. Vastaanotettu dataobjekti on tulostettavissa käyttäjän tarkasteltavaksi asiakkaan yhteydessä olevan käyttöliittymän välityksellä.

Kuvion 2 lohkokaavion havainnollistaa asiakkaana käytettävää langatonta päätelaitetta. Esitetyssä suoritusmuodossa päätelaitteena on käytetty matkaviestintä keksintöä käytettyyn laitetyyppiin tai termeihin rajoittamatta. Päätelaite voi olla mikä tahansa langaton tietoliikenneväline, kuten esimerkiksi kaksisuuntainen hakulaite, langaton PDA-laite (Personal Digital Assistant), IP-protokollaa käyttävä WLAN-päätelaite (Wireless Local Area Network) tai kannettava tietokone, joka on varustettu laitteistoporttiin lisättävällä antennin

käsittävällä matkaviestinverkkokortilla. Kuvion 2 lohkokaaviossa kuvattu matkaviestin sisältää radioteitse tapahtuvaa kommunikointia varten radioyksikön, joka käsittää tavanomaisesta matkaviestimestä tunnetun lähetinhaaran (käsittäen kanavakoodauksen, lomituksen, salauksen, moduloinnin ja lähetyksen suorittavat toimintalohkot) 21, vastaanotinhaaran (käsittäen vastaanoton, demoduloinnin, salauksen purun, lomituksen purun sekä kanavadekoodauksen suorittavat toimintolohkot) 22, radioteitse tapahtuvaa lähetystä varten vastaanoton ja lähetyksen erottavan duplex-suodattimen 23 ja antennin 24. Päätelaitteen toimintaa kokonaisuudessaan ohjaa keskusyksikkö 25, joka myös toteuttaa päätelaitteen yhteyskäytännön mukaiset toiminnallisuudet. Matkaviestin käsittää muistin 26, joka sisältää haihtuvaa ja haihtumatonta muistia, ja liityntäyksikön 27, joka käsittää yhden tai useampia laitteistoportteja sisäisten tai ulkoisten lisälaitteiden liittämiseksi matkaviestimeen. Käyttäjän kanssa suoritettavaa kommunikointia varten päätelaite käsittää käyttöliittymän, joka sisältää tyypillisesti näppäimistön, näytön, mikrofonin ja kaiuttimen. Älykorttisovellusten yhteydessä liityntäyksikkö 27 käsittää kortinlukijan, jonka välityksellä keskusyksikkö 25 kommunikoi lukijaan asetetun kortin kanssa. Yhteys palvelimeen toteutetaan radioyksikön 21, 22, 23, 24 välityksellä. Keskusyksikkö 25 ohjaa älykorttisovelluksen toteutusta suorittamalla laitteen muistiin 26 ohjelmallisesti tai laitteistorakentein järjestetyt toiminnot, ja edullisesti palvelimelta päätelaitteeseen ladatun sovelluskoodin toiminnot.

Kuviossa 2 esitettiin kuvion 1 mukaisena asiakkaana toimivan langattoman päätelaitteen rakenteelliset yksiköt, erityisesti älykorttisovellusten kannalta. Kuviossa 3 esitellään vastaavasti keksinnön mukaisen ratkaisun toiminnalliset elementit. Kuvion 3 asiakas muodostuu langattomasta päätelaitteesta 10 ja älykortista 15. Älykorttisovelluksen käynnistyttyä päätelaitteelle 10 ladataan palvelimelta sovelluskoodi 31. Päätelaite 10 käsittää ajurin 32, joka ladatun sovelluskoodin käynnistämänä kutsuu laitteeseen tallennetusta koodikirjastosta 33 funktioita, jotka toteuttavat määrätyn tiedonsiirtoyhteyden määrätyn laiteyksikön, kuvion 3 esimerkissä älykortin 15, kanssa. Tiedonsiirtoyhteys toteutetaan

funktioiden sisältämien, sovellusyhteyskäytännön mukaisten tietoyksiköiden (Application Protocol Data Unit, APDU) välityksellä.

Kuviossa 4 on esitetty tarkemmin keksinnön mukainen järjestely älykorttisovelluksen toteuttamiseksi langattomassa päätelaitteessa. Päätelaitteelle on ladattu yksi tai useampia ajureita 41, 42, 43, jotka on tallennettu päätelaitteelle tavukoodatussa muodossa. Tavukäännetty koodi tarkoittaa tässä yhteydessä lähdekoodista käännettyä ohjelmaa, jonka käskyt ovat suoritettavissa päätelaitteelle järjestetyllä virtuaalikoneella. Virtuaalikone tässä yhteydessä tarkoittaa ohjelmistoa, joka sisältyy laitteen järjestelmäalustaan, ja joka mahdollistaa virtuaalikoneen tukemalla kielellä toteutetun, laitteelle ladatun tavukäännetyn koodin suorittamisen laitteessa. Tavukäännetyn koodin etuna on, että sovellusohjelmia voidaan kehittää huolehtimatta siitä, mikä sovelluksen käyttöympäristö on. Koodi on vakiomuotoista, joten sovellusten käyttäjäpäässä on vain huolehdittava, että oikea virtuaalikone sisältyy laitteen järjestelmäalustaan.

Tunnetuin esimerkki tavukäännetystä koodista on www-ympäristöstä tunnettu Java. WAP-sovelluskerros WAE (Wireless Application Environment) tarjoaa rajoitettuun ympäristöön soveltuvan mikroselaimen, jossa tiedon kuvauskieli on WML (Wireless Markup Language), ja ohjelmointikieli WMLScript. WMLScript on joiltakin ominaisuuksiltaan JavaScript-ohjelmointikielen kaltainen, mutta WMLScript on optimoitu toimimaan rajoitettua kaistanleveyttä käyttävissä muisti- ja laskentakapasiteetiltaan rajoitetuissa päätelaitteissa.

Ajurit 41, 42, 43 tarjoavat vakiomuotoisen ylemmän tason rajapinnan (High Level Application Programming Interface, HL-API) päätelaitteeseen käyttövaiheessa ladattavalle sovelluskoodille. Edullisesti samaan käyttötarkoituksiin tarkoitetuille eri sovelluksille määritetään yhteinen rajapinta, jolloin sovellusta ohjelmoivan osapuolen ei tarvitse välttämättä huolehtia sovellusten välisistä eroista, ja ohjelmointi helpottuu. Esimerkiksi kuviossa 4 rajapinta 44 on elektroniseen kukkaroon liittyvien sovellusten rajapinta, rajapinta 45 on

asiakaspalkkiosovelluksiin liittyvä rajapinta ja rajapinta 46 on potilastietosovelluksiin liittyvä rajapinta.

Päätelaitteeseen 10 tallennettu koodikirjasto käsittää kaksi osaa, vakiokirjaston 41 ja suljetun kirjaston 42. Koodikirjastot 41, 42 käsittävät edullisesti yhden tai useampia konekielisessä muodossa tallennettuja funktioita. Konekielinen koodi tässä yhteydessä tarkoittaa lähdekielisestä ohjelmasta kääntäjällä käsiteltyä tiedostoa, joka on valmis suoritettavaksi laitteen suorittimella laitteen käyttöjärjestelmän kutsusta. Keksinnön mukaisen vakiokirjaston 41 funktiot ovat perusfunktioita (esimerkiksi käyttöliittymän toiminnot, matemaattiset funktiot), joita voi kutsua mikä tahansa päätelaitteeseen tallennettu tai ladattu ajuri.

Suljettu kirjasto 48 käsittää funktioita, jotka mahdollistavat läpinäkyvän tiedonsiirron sovelluskoodin ja älykortin 15 välillä. Tällainen funktio voi olla esimerkiksi *sendAPDU(CommandAPDU)*, jolle annetaan parametriksi sovelluskoodin HL-API-käskyn mukaisen ajurin sisältämä APDU. Ajurin ja älykortin välillä on siis geneerinen alemman tason rajapinta (Low Level Application Programming Interface, LL-API), ja oleellisesti mitkä tahansa käynnistettyyn sovellukseen liittyvät sovelluskoodin komennot välittyvät läpinäkyvästi ajurin läpi älykortille ja päinvastoin. HL-API helpottaa sovelluskoodin laatimista, koska sovelluksen laatijan ei tarvitse välttämättä tuntea yksittäisten ratkaisujen detaljeja; LL-API vastaavasti mahdollistaa ajurista riippumattoman läpinäkyvän tiedonsiirtoyhteyden ajurin ja älykortin välillä.

On kuitenkin selvää, että esitetyn järjestelyn geneerisyys tällaisenaan sisältää potentiaalisen riskin epärehellisten sovellustarjoajien tai sovelluskoodin laatijoiden suhteen. Sovelluskoodin kirjoittaja voisi esimerkiksi sisällyttää koodiin käskyn siirtää jokaisen kukkarotransaktion yhteydessä pienen summan rahaa omalle tililleen. Turvallisuusriskin välttämiseksi keksinnön mukaisessa järjestelyssä suljetun kirjaston funktioita voi kutsua vain ajuri, jonka alkuperä on elektronisen allekirjoituksen avulla varmistettu.

Elektronisessa allekirjoituksessa hyödynnetään julkisen avaimen salaustekniikkaa, esimerkiksi RSA-algoritmia., lisäksi WAP-ympäristössä elektronisissa allekirjoituksissa hyödynnetään muun muassa elliptisiä käyriä. Eri allekirjoitustekniikoin toteutetut ratkaisut ovat keksinnön kannalta ekvivalentteja. Seuraavassa esitetään kuvion 5 toimenpidekaavion avulla eräs mahdollinen suoritusmuoto. Allekirjoitettava viesti 51, eli tässä tapauksessa palvelimella oleva ajurikoodi käsitellään tiivistefunktiolla 52, joka määrätyn algoritmin (esimerkiksi MD5) avulla laskee viestistä määrämittaisen tiivisteen 53. Tiivistefunktio 52 on yksisuuntainen funktio, jonka tuottama tiiviste 53 muuttuu heti, jos alkuperäistä viestiä 51 muutetaan vähänkin. Tiiviste 53 käsitellään salausfunktiolla 54 käyttäen salausavaimena ajurin julkistajan salaista avainta. Toimenpiteen tulos on elektroninen allekirjoitus 55, joka liitetään ajurikoodiin. Kun ajuri halutaan verifioida, allekirjoitus ensin avataan käyttäen ajurin julkistajan julkista avainta. Tämän jälkeen ajurikoodi käsitellään tiivistefunktiolla. Jos mainittu julkinen avain avaa allekirjoituksen ja avattu merkkijono vastaa laskettua tiivistettä, tiedetään, että ajuri on peräisin halutusta lähteestä ja ettei koodia ole käsitelty julkistuksen jälkeen. Luottamusketju edellyttää olennaisesti julkisen avaimen siirtoa, ja aikaansaatu turvallisuustaso on erittäin hyvä.

Kuviossa 6 on havainnollistettu yksityiskohtaisemmin järjestely yhteen käyttötarkoitukseen tarkoitettujen sovellusten toteuttamiseksi päätelaitteella. Kuvion 6 tapauksessa kyseessä on elektroninen kukkaro, jossa jokaiselle kukkarosovelluksen tarjoajalle on edullisesti tarjoajan itsensä määrittämä ajuri 61, 62, 63, joka voidaan ladata päätelaitteelle tarvittaessa palvelimelta. Maksutapahtumaan liittyvä sovelluskoodi voi kuitenkin HL-APIn ansiosta olla sama kaikille eri kukkarosovelluksille. Ladattu ajuri, jonka oikeellisuus on varmistettu elektronisen allekirjoituksen avulla, joko latauksen yhteydessä tai maksutapahtuman yhteydessä, pystyy kutsumaan sekä vakiokirjaston että suljetun kirjaston funktioita, ja siten ohjata maksutapahtuman suoritusta turvallisuusvaatimukset ylläpitäen maksupalvelimen ja älykortin välillä.

Seuraavassa tarkastellaan esimerkin avulla ajurin latausta päätelaitteelle. Käyttäjä on tehnyt sopimuksen älykorttiperusteista maksusovellusta tarjoavan yrityksen X kanssa. Yritys antaa käyttäjälle älykortin, joka sisältää elektronisen kukkarosovelluksen XCash. Saadakseen kyseiseen XCash-sovellukseen liittyvän ajurin päätelaitteeseensa käyttäjä esimerkiksi:

a) aktivoi WAP-puhelimensa, johon on liitetty ulkoinen älykortin lukija. Hän lähettää yrityksen X WAP-palvelimelle (esimerkiksi valitsemalla sopivan linkin) pyynnön ladata XCash-tuki WAP-puhelimeensa. Tai

b) työntää uuden XCash-kortin WAP-puhelimen kortinlukijaan. Kortinlukija eli päätelaite lähettää älykortille nollaussignaalin (Reset), ja nollauksen jälkeen älykortti vastaa lähettämällä nollausvastaussekvenssin (Answer To Reset, ATR) kortinlukijalle. ATR-vastauksessa saadun tiedon perusteella kortinlukija päättelee, että kyseessä on uusi maksukorttisovellus, ja käynnistää automaattisesti ajurin latauksen palvelimelta. Voidakseen käynnistää latauksen, päätelaite tarvitsee osoitetiedon paikallistaakseen sen palvelimen, josta ajuri on ladattavissa. Osoite (esim. Universal Resource Locator, URL) voi olla tallennettuna esimerkiksi älykorttiin itseensä tai päätelaitteeseen. Eräs mahdollisuus on järjestää verkkoon palvelin, jolla ylläpidetään ajureita erilaisia älykorttisovelluksia varten, tai jonne tallennetaan tieto (esim. URL), jonka perusteella päätelaite löytää oikean ajurin.

WAP-yhteyskäytäntöä hyödyntäen käyttäjä lähettää palvelimelle pyynnön lähettää valittu ajuri WAP-puhelimelle. Palvelin lähettää ajurin, joka sisältää elektronisen allekirjoituksen. Elektroninen allekirjoitus voidaan tarkistaa tässä vaiheessa ennen ajurin tallennusta päätelaitteen haihtumattomaan muistiin (esimerkiksi välimuisti, cache). Jos muistitila on täyttymässä, käyttäjää voidaan pyytää poistamaan vanhempia, käyttämättömäksi jääneitä ajureita.

Seuraavassa tarkastellaan esimerkin avulla ladatun ajurin käyttöä päätelaitteella. Käyttäjä selaa WAP-palvelimella olevaa sisältöä WAP-puhelimellaan ja kiinnostuu yrityksen Y tarjoamasta maksullisesta WML-asiakirjasta. Ilmaistuaan halunsa maksaa kyseisen asiakirjan XCash-palveluna palvelin siirtää WMLScript-sovelluskoodin käyttäjän WAP-puhelimeen. WAP-puhelimen järjestelmäalusta

sisältää WMLScript virtuaalikoneen, joka alkaa suorittaa ladattua WMLScript-sovelluskoodia. Käynnistääkseen EUR 10 suuruisen summan siirron sovelluskoodi antaa HL-API-käskyn

EPurse.StartPayment("XCash", "EUR", 10, "serverAddress")

joka aikaansaa päätelaitteelle tallennetun XCash-ajurin käynnistymisen. Ajuri ohjaa maksutapahtumia päätelaitteella kutsuen esimerkiksi vakiokirjaston funktiota kysymään käyttäjäliitynnän välityksellä käyttäjän valtuutusta maksulle ja valtuutuksen saatuaan antaa sopivan APDUn XCash-kortille suljetun kirjaston funktion avulla. Vastaavasti ajuri ohjaa WAP-selaimen avulla yhteyttä yrityksen Y ja mahdollisesti myös yrityksen X maksupalvelimeen hoitaen maksutapahtumaa verkon suuntaan tunnetulla tavalla. Kun yritys Y on saanut tiedon siitä, että maksutapahtuma on päättynyt, se siirtää halutun asiakirjan verkon yli käyttäjän WAP-puhelimeen.

Edellä esitetyissä suoritusmuodoissa LL-API on rajapinta, jonka läpi voidaan lähettää APDUja päätelaitteeseen liitettävälle prosessointivälineelle, tässä älykortille. Keksinnön mukainen ratkaisu on kuitenkin sovellettavissa myös muihin ympäristöihin, joissa yhteys sovellusohjelmasta sovellusta toteuttavalle suoritusvälineelle on aikaansaatava joustavasti mutta samalla riittävä turvallisuustaso ylläpitäen. Suoritusväline voi olla verkkoelementtiin liitetty tai verkkoelementin sisäinen suoritin. LL-API voi vastaavasti olla esimerkiksi Socket API, jonka avulla avataan yhteys sovellusohjelman ja verkossa olevan palvelimen välille. Ajurin lataaminen verkosta mahdollistaa joustavuuden ja nopeuden uusien sovellusten yhteydessä ja LL-API takaa, että yhteys on toteutettavissa vain ajurilla, jonka alkuperä on varmistettavissa. Eräs toinen suoritusmuoto on päätelaitteeseen liitettävän GPS-paikannuslaitteen (Global Positioning System) LL-API, joka mahdollistaa paikannusta hyödyntävien sovellusten joustavan käyttöönoton säilyttäen samalla mahdollisuuden kontrolloida paikannustietoa saavan sovellusohjelman alkuperän.

Tässä on esitetty keksinnön toteutusta ja suoritusmuotoja esimerkkien avulla. Alan ammattimiehelle on ilmeistä, ettei keksintö rajoitu edellä esitettyjen

suoritusmuotojen yksityiskohtiin ja että keksintö voidaan toteuttaa muussakin muodossa poikkeamatta keksinnön tunnusmerkeistä. Näinollen esimerkiksi päätelaite voi olla muukin kuin edellä esitetty WAP-puhelin. Siten keksinnön toteutus- ja käyttömahdollisuuksia rajoittavatkin ainoastaan oheistetut patenttivaatimukset, ja vaatimusten määrittelemät erilaiset keksinnön toteutusvaihtoehdot, myös ekvivalenttiset toteutukset kuuluvat keksinnön piiriin.

Patenttivaatimukset

1. Tiedonsiirtoverkon verkkoelementti, joka käsittää

tiedonsiirtovälineet (21, 22, 23, 24) tiedon lähettämiseksi ja vastaanottamiseksi tiedonsiirtoverkosta, joka tieto käsittää yhden tai useampia suorituskäskyjä;

käsittelyvälineet (15, 25) määrätyssä muodossa annetun tiedon käsittelemiseksi;

ohjausvälineet (25, 27, 28, 29) vastaanotetun yhden tai useamman suorituskäskyn muokkaamiseksi käsittelyvälineiden edellyttämään muotoon;

tunnettu siitä, että mainitut ohjausvälineet käsittävät

ajurin (41, 42, 43), jonka alkuperä on varmennettavissa elektronisen allekirjoituksen avulla;

yhden tai useampia käsittelyvälineiden (15, 25) toimintaa ohjaavia funktioita (48), jotka ovat käynnistettävissä vain ajurilla (41, 42, 43), jonka alkuperä on varmennettu elektronisen allekirjoituksen avulla.

2. Patenttivaatimuksen 1 mukainen verkkoelementti, **tunnettu** siitä, että mainittu verkkoelementti on langaton päätelaite.

3. Patenttivaatimuksen 2 mukainen verkkoelementti, **tunnettu** siitä, että mainittu päätelaite sisältää välineet (28) sovellusprosessointivälineen (15) liittämiseksi päätelaitteeseen.

4. Patenttivaatimuksen 3 mukainen verkkoelementti, **tunnettu** siitä, että mainitut käsittelyvälineet käsittävät sovellusprosessointivälineen (15).

5. Patenttivaatimuksen 1 mukainen verkkoelementti, **tunnettu** siitä, että mainittu ajuri (41, 42, 43) säilytetään verkkoelementissä tavukäännettynä koodina.

6. Patenttivaatimuksen 5 mukainen verkkoelementti, **tunnettu** siitä, että mainittu elektroninen allekirjoitus käsittää mainitun koodin julkaisijan salaisella avaimella

käsitellyn merkkijonon, jolloin ajurin alkuperän varmennus tapahtuu purkamalla merkkijonon salaus koodin julkaisijan julkisella avaimella.

7. Patenttivaatimuksen 6 mukainen verkkoelementti, **tunnettu** siitä, että mainittu merkkijono käsittää mainitusta koodista määrätyllä tiivistefunktiolla lasketun tiivisteen.

8. Patenttivaatimuksen 1 mukainen verkkoelementti, **tunnettu** siitä, että määrättyyn käyttötarkoitukseen liittyvä ajuri (61) on järjestetty vastaanottamaan suorituskäsky määrätyssä, käyttötarkoituksen mukaan määräytyvässä muodossa.

9. Tiedonsiirtoverkko, jonka verkkoelementti (10, 5) käsittää

tiedonsiirtovälineet (21, 22, 23, 24) tiedon lähettämiseksi ja vastaanottamiseksi tiedonsiirtoverkosta, joka tieto käsittää yhden tai useampia suorituskäskyjä;

käsittelyvälineet (15, 25) määrätyssä muodossa annetun tiedon käsittelemiseksi;

ohjausvälineet (25, 27, 28, 29) vastaanotetun suorituskäskyn muokkaamiseksi käsittelyvälineiden edellyttämään muotoon;

tunnettu siitä, että mainitut ohjausvälineet käsittävät

ajurin (41, 42, 43), jonka alkuperä on varmennettavissa elektronisen allekirjoituksen avulla;

yhden tai useampia käsittelyvälineiden (15, 25) toimintaa ohjaavia funktioita (48), jotka ovat käynnistettävissä vain ajurilla (41, 42, 43), jonka alkuperä on varmennettu elektronisen allekirjoituksen avulla.

10. Patenttivaatimuksen 9 mukainen tiedonsiirtoverkko, **tunnettu** siitä, että mainittu verkkoelementti on langaton päätelaite (10).

11. Patenttivaatimuksen 9 mukainen tiedonsiirtoverkko, **tunnettu** siitä, että mainittu verkkoelementti on verkkopalvelin (5).

12. Menetelmä tiedon käsittelemiseksi, jossa menetelmässä:

lähetetään ja vastaanotetaan tiedonsiirtoverkosta tieto, joka käsittää yhden tai useampia suorituskäskyjä;

muokataan vastaanotettu suorituskäsky määrättyyn muotoon;

käsitellään määrätyssä muodossa oleva tieto;

tunnettu siitä, että

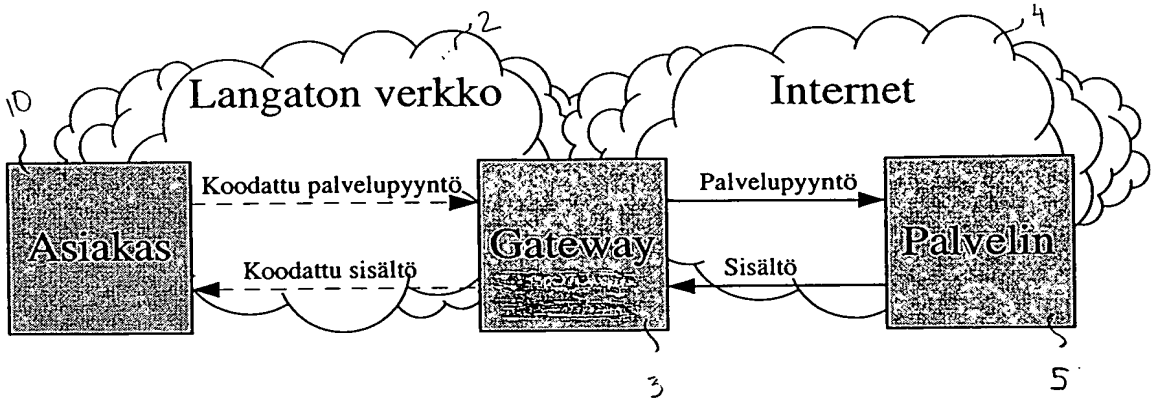
muokataan suorituskäsky ajurilla, jonka alkuperä on varmennettavissa elektronisen allekirjoituksen avulla;

ohjataan käsittelyä funktioilla, jotka ovat käynnistettävissä vain ajurilla, jonka alkuperä on varmennettu elektronisen allekirjoituksen avulla.

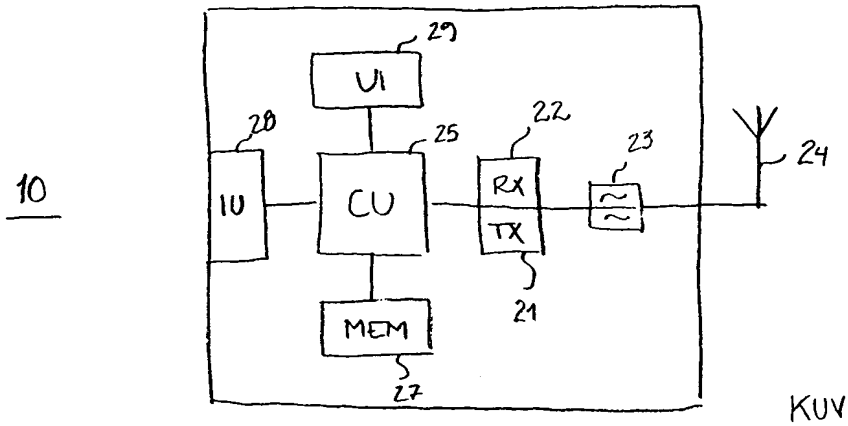
(57) Tiivistelmä

Ratkaisu verkkoelementissä suoritettavien toimenpiteiden ohjaamiseksi joustavalla ja turvallisella tavalla. Tiedonsiirtoverkko käsittää verkkoelementin, joka käsittää tiedonsiirtovälineet (21, 22, 23, 24) tiedon lähettämiseksi ja vastaanottamiseksi tiedonsiirtoverkosta, joka tieto käsittää yhden tai useampia suorituskäskyjä; käsittelyvälineet (15, 25) määrätyssä muodossa annetun tiedon käsittelemiseksi; ja ohjausvälineet (25, 27, 28, 29) vastaanotetun suorituskäskyn muokkaamiseksi käsittelyvälineiden edellyttämään muotoon. Mainitut ohjausvälineet käsittävät ajurin (41, 42, 43), jonka alkuperä on varmennettavissa elektronisen allekirjoituksen avulla; ja yhden tai useampia käsittelyvälineiden (15, 25) toimintaa ohjaavia funktioita (48), jotka ovat käynnistettävissä vain ajurilla (41, 42, 43), jonka alkuperä on varmennettu elektronisen allekirjoituksen. avulla.

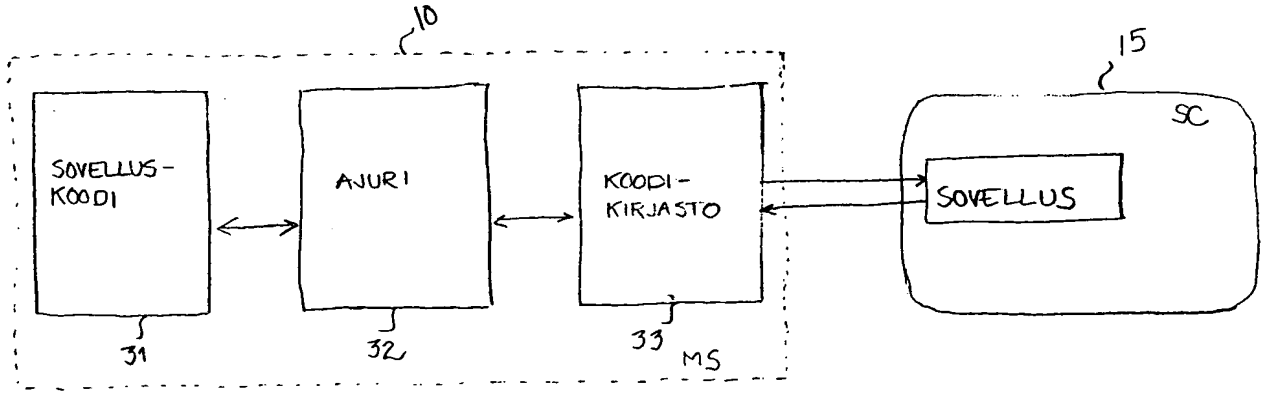
(Kuvio 4)



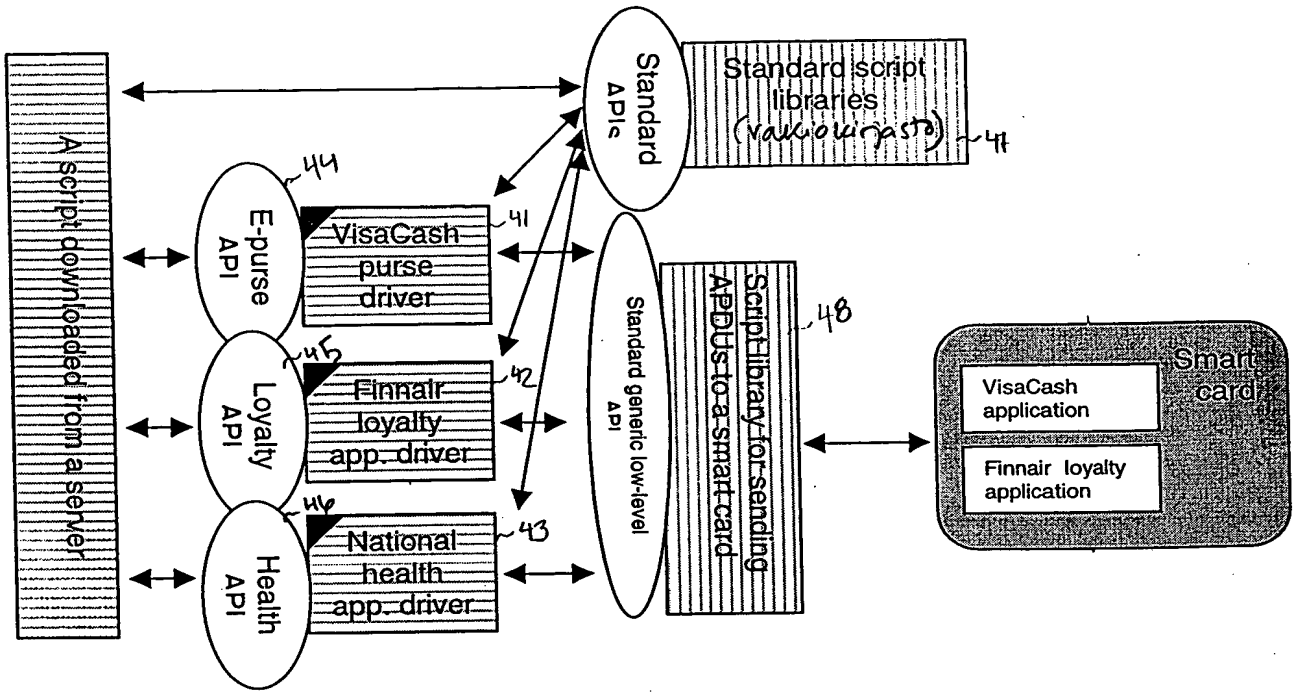
KUVIO 1



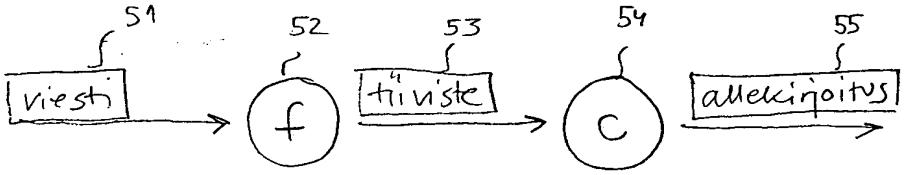
KUVIO 2



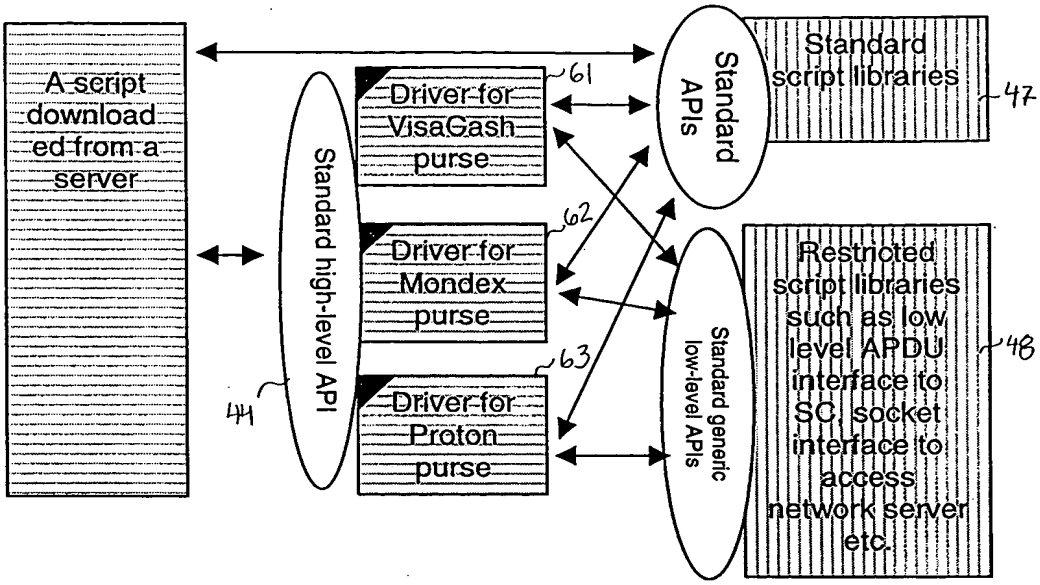
KUVIO 3



KUVIO 4



KUVIO 5


A script downloaded from a server
Standard high-level API
44
Driver for VisaCash purse
61
Driver for Mondex purse
62
Driver for Proton purse
63
Standard APIs
Standard script libraries
47
Standard generic low-level APIs
Restricted script libraries such as low level APDU interface to SC socket interface to access network server etc.
48

KUV106